

PATENT

Docket No. RSW920010081US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTORS: David B. Colasurdo et al.

APPLICATION NO. 09/845,502

Confirmation Number: 2095

FILED: April 30, 2001

Examiner: Blair, D.

CASE NO. RSW920010081US1

Group Art Unit: 2142

TITLE: METHOD AND APPARATUS FOR MAINTAINING SESSION
AFFINITY ACROSS MULTIPLE SERVER GROUPS

FILED ELECTRONICALLY ON July 2, 2007

MAIL STOP APPEAL BRIEF-PATENTS

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Attention: Board of Patent Appeals and Interferences

APPELLANTS' BRIEF

This Appeal Brief is in furtherance of the Notice of Appeal filed in this case on May 2, 2007. The Commissioner is authorized to charge the fee for filing of this Appeal Brief to Deposit Account No. 09-0461.

1. REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, having its principal place of business at New Orchard Road, Armonk, New York 10504. Accordingly, International Business Machines Corporation is the real party in interest.

2. RELATED APPEALS AND INTERFERENCES

The Appellants, assignee, and the legal representatives of both are unaware of any other appeal or interference which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

3. STATUS OF CLAIMS

- A. Claims canceled: None
- B. Claims withdrawn from consideration but not canceled: None
- C. Claims pending: 1-29
- D. Claims allowed: none
- E. Claims rejected: 1-29
- F. Claims appealed: 1-29

Appealed claims 1-29 as currently pending are attached as the Claims Appendix hereto.

4. STATUS OF AMENDMENTS

A Reply under 37 C.F.R. §1.111 was filed on August 31, 2005; claim amendments were made. In response, the Examiner entered the claim amendments and issued the final Office Action appealed herein on February 2, 2007.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER

Claim 1: A method of maintaining session affinity in a server farm coupled to receive client requests, said server farm comprising multiple server groups, each server group comprising multiple clone servers, said method comprising the steps of: (1) associating a collection of related client requests with a unique session identification code (page 16, lines 5-19); (2) responsive to receipt of a client request, determining to which of said server groups said client request can be dispatched (page 16, line 19 – page 17, line 6); (3) associating with said collection of requests sharing a session identification code a list of every server in said server farm that has serviced a request in said collection (page 17, lines 7-20); (4) responsive to said receipt of a client request which comprises a session identification code, determining if said list associated with said received session identification code includes a server identification code that matches a server identification code of a server in said determined server group (page 17, lines 7-20); and (5) if a match is detected, dispatching said client request to said matched server (page 19, lines 5-9).

Claim 16: A computer readable product embodied on computer readable media

readable by a computing device, said product for maintaining session affinity in a server farm coupled to receive client requests, said server farm comprising multiple server groups, each server group comprising multiple server clones, said computer program product comprising computer executable instructions for: associating a collection of related client requests with a unique session identification code (page 16, lines 5-19); responsive to receipt of a client request, determining to which of said server groups said client request can be dispatched (page 16, line 19 – page 17, line 6); associating with said collection of requests sharing a session identification code a list of every server in said server farm that has serviced a request in said collection (page 17, lines 7-20); responsive to said receipt of a client request which comprises a session identification code, determining if said list associated with said received session identification code includes a server identification code that matches a server identification code of a server in said determined server group (page 17, lines 7-20); and if a match is detected, dispatching said client request to said matched server (page 19, lines 5-9).

The present invention relates to a method and apparatus of using information pertaining to previous client requests which are grouped under a session identification code to maintain session affinity. This information includes a list of servers that previously serviced those requests. For additional requests, the session identification code of the newly received request is utilized to assign an available server which has processed a related client request.

6. **GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

Appellants request the Board to review the following rejections:

1. Rejection of Claims 1-29 under 35 U.S.C. §103(a) based on U.S. Patent No. 6,598,077 to Primak et al. in view of U.S. Patent Application Publication No. 2002/0073211 to Lin et al.

7. **ARGUMENT**

The Cited Art Does Not Render the Claims Obvious

The Examiner Has Not Established a *Prima Facie* Case of Obviousness

As set forth in the MPEP:

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP 2143

As briefly discussed above, present invention maintains server affinity by grouping together client requests under a session identification code. These previous requests are linked to the servers that handled the requests by the session identification codes. When additional requests are received from previously served clients, the clients attach the session identification code from any previous requests. This enables the request to be quickly forwarded to a server that has previously handled this request. One important aspect of the present invention is associating with a collection of requests

sharing a session identification code a list of every server in the server group that has serviced a request in the collection. This is important as it provides a means for load balancing without losing session affinity. If the server that previously handled a client request is busy and cannot handle a new request from the same client, a server that has handled a similar request from the same group can handle the client's new request and affinity is maintained. This feature is not taught or reasonably suggested by the prior art of record and renders the present invention as non-obvious.

Primak teaches a system for directing a request for dynamic content to an application server having access to the dynamic content. However, Primak is not concerned with session affinity; rather Primak is concerned with matching a user to a server which is capable of providing the requested content. As acknowledged by the Examiner, Primak fails to teach associating with a collection of requests sharing a session identification code a list of every server in the server group that has serviced a request in the collection as Primak is not concerned with session affinity.

Next the Examiner looks to Lin to teach associating with a collection of requests sharing a session identification code a list of every server in the server group that has serviced a request in the collection. However, Lin teaches a system for providing secure communications between application servers and a series of users. Figure 6 and the accompanying text (specifically paragraphs [0047] – [0051]) provide an accurate overview of the system of Lin. A series of application servers are connected to a series of approved web servers. The series of web servers are connected to a load balancer that coordinates user requests and distributes the requests to various web

servers to assure no individual server gets overloaded. However, load balancing as taught by Lin is not the same as maintaining session affinity as is the goal of the present invention. Specifically, Lin fails to teach associating with a collection of requests sharing a session identification code a list of every server in the server group that has serviced a request in the collection as is asserted by the Examiner. Rather, Lin teaches maintaining a record of transactions at a central server. But this information is not utilized to assure a user with a specific session identification code is connected to the same server again. For example, Figure 7 of Lin outlines the operation of the invention. Step 704 is "Load Balancer Routes Browser to Available Webserver." This step is performed before any session identification or previous activity information is examined. In contrast to the present invention, this load balancing of Lin prevents session affinity.

Without such a teaching or suggestion, the combination of Primak and Lin fails to render the present invention non-obvious and Appellants requests this board to reconsider and withdraw the rejections of Claims 1-29 under 35 U.S.C. 103(a).

8. CONCLUSION

For the foregoing reasons Appellants respectfully request this Board to overrule the Examiner's rejections and allow Claims 1-29.

Respectfully submitted,

July 2, 2007
Date

/John R. Brancolini/
John R. Brancolini, Reg. No. 57,218
Synnestvedt & Lechner LLP
1101 Market Street
Suite 2600
Philadelphia, PA 19107
Telephone: 215-923-4466
Facsimile: 215-923-2189

CLAIMS APPENDIX

CLAIMS INVOLVED IN THIS APPEAL:

1. (Previously presented): A method of maintaining session affinity in a server farm coupled to receive client requests, said server farm comprising multiple server groups, each server group comprising multiple clone servers, said method comprising the steps of:

(1) associating a collection of related client requests with a unique session identification code;

(2) responsive to receipt of a client request, determining to which of said server groups said client request can be dispatched;

(3) associating with said collection of requests sharing a session identification code a list of every server in said server farm that has serviced a request in said collection;

(4) responsive to said receipt of a client request which comprises a session identification code, determining if said list associated with said received session identification code includes a server identification code that matches a server identification code of a server in said determined server group; and

(5) if a match is detected, dispatching said client request to said matched server.

2. (previously presented): The method of claim 1 wherein step (3) comprises, upon routing of a client request in a session to a server that has not previously serviced

a client request in said session, adding a unique server identification code corresponding to said server to a list of server identification codes associated with said session, without deleting any other server identification codes in said list.

3. (previously presented): The method of claim 2 wherein steps (1) and (3) comprise sending said session identification code and said unique server identification code, respectively, to a client machine that issued said request so that said client machine can include said session identification code and said unique server identification code as part of future requests it issues to said server farm.

4. (original): The method of claim 3 wherein said list of server identification codes forms part of said session identification code.

5. (original): The method of claim 4 wherein said list of server identification codes is appended to said session identification code.

6. (original): The method of claim 5 wherein said session identification code is a `jsessionid` in accordance with the Java Servlet 2.2 specification.

7. (original): The method of claim 5 wherein said session identification code forms part of a cookie associated with said session.

8. (original): The method of claim 7 wherein said cookie forms part of said client request.

9. (original): The method of claim 5 wherein said session identification code forms part of a Uniform Resource Identifier (URI) that forms part of said client request.

10. (original): The method of claim 4 wherein step (4) comprises the steps of:
(4.1) parsing said list in a manner so as to find matching server identification codes in the temporal order in which they were added to said list; and
(4.2) selecting the first matching server identification code detected.

11. (previously presented): The method of claim 1 wherein step (4) comprises the steps of:
(4.1) searching said list in a predetermined order for matching server identification code, said predetermined order designed to encounter server identification codes in the temporal order in which they were added to said list;
(4.2) upon encountering a server identification code, determining if it matches a server identification code in said server group; and
(4.3) if a match is not detected, repeating steps (4.1) and (4.2).

12. (original): The method of claim 4 wherein step (3) comprises the steps of:
if said client request does not comprise a session identification code;

(3.1) creating a session for client requests received from the client issuing said client request;

(3.2) assigning a session identification code to said session;

(3.3) dispatching said client request to a particular server in said determined server group;

(3.4) appending the server identification code of said particular server to said session identification code; and

(3.5) communicating said session identification code with said server identification code to the client that issued said client request.

13. (original): The method of claim 12 wherein step (3) further comprises the steps of:

when a client request is received that pertains to a particular server group and said client request comprises a session identification code, but said session identification code does not have appended thereto a server identification code corresponding to a server in said particular server group;

(3.6) dispatching said client request to a server in said particular server group;

(3.7) appending said server identification code corresponding to said server to said session identification code; and

(3.8) communicating said session identification code, including said newly appended server identification code, to the client that issued said client request.

14. (original): The method of claim 13 wherein step (2) further comprises the steps of:

if a client request comprises a session identification code having appended thereto a server identification code corresponding to a particular server in said determined server group, but said particular server is unavailable to service said client request;

(3.9) dispatching said client request to a different server in said determined server group;

(3.10) appending a server identification code corresponding to said different server to said session identification code; and

(3.11) communicating said session identification code, including said newly appended server identification code, to the client that issued said client request.

15. (original): The method of claim 5 wherein step (3) further comprises the steps of:

if a client request comprises a session identification code having appended thereto a server identification code corresponding to a particular server in said determined server group, but said particular server is unavailable to service said client request;

(3.9) dispatching said client request to a different server in said determined server group;

(3.10) appending a server identification code corresponding to said different server to said session identification code; and

(3.11) communicating said session identification code, including said newly appended server identification code, to the client that issued said client request.

16. (Previously presented): A computer readable product embodied on computer readable media readable by a computing device, said product for maintaining session affinity in a server farm coupled to receive client requests, said server farm comprising multiple server groups, each server group comprising multiple server clones, said computer program product comprising computer executable instructions for:

associating a collection of related client requests with a unique session identification code;

responsive to receipt of a client request, determining to which of said server groups said client request can be dispatched;

associating with said collection of requests sharing a session identification code a list of every server in said server farm that has serviced a request in said collection;

responsive to said receipt of a client request_which comprises a session identification code, determining if said list associated with said received session identification code includes a server identification code that matches a server identification code of a server in said determined server group; and

if a match is detected, dispatching said client request to said matched server.

17. (previously presented): The method of claim 16 wherein said computer executable instructions for associating said list with said collection of requests comprises, upon routing of a client request in a session to a server that has not previously serviced a client request in said session, adding a unique server identification code corresponding to said server to a list of server identification codes associated with said session, without deleting any other server identification codes in said list.

18. (previously presented): The method of claim 17 wherein said computer executable instructions for associating a collection of related client requests with a unique session identification code and said computer executable instructions for associating said list with said collection of requests comprise sending said session identification code and said unique server identification code, respectively, to a client machine that issued said request so that said client machine can include said session identification code and said unique server identification code as part of future requests it issues to said server farm.

19. (original): The computer readable product of claim 18 wherein said list of server identification codes forms part of said session identification code.

20. (original): The computer readable product of claim 19 wherein said list of server identification codes is appended to said session identification code.

21. (original): The computer readable product of claim 19 wherein said session identification code is a jsessionid in accordance with the Java Servlet 2.2 specification.

22. (original): The computer readable product of claim 19 wherein said session identification code forms part of a cookie associated with said session.

23. (original): The computer readable product of claim 22 wherein said cookie forms part of said client request.

24. (original): The computer readable product of claim 19 wherein said session identification code forms part of a Uniform Resource Identifier (URI) that forms part of said client request.

25. (original): The computer readable product of claim 18 wherein said computer executable instructions for determining if said list associated with said collection of requests to which said request belongs includes a server identification code that matches a server identification code of a server in said determined server group comprises computer executable instructions for:

parsing said list in a manner so as to find matching server identification codes in the temporal order in which they were added to said list; and

selecting the first matching server identification code detected.

26. (original): The computer readable product of claim 18 wherein said computer executable instructions for determining if said list includes a server identification code that matches a server identification code of a server in said determined server group comprises computer executable instructions for:

searching said list in a predetermined order for matching server identification code, said predetermined order designed to encounter server identification codes in the temporal order in which they were added to said list;

upon encountering a server identification code, determining if it matches a server identification code in said server group; and

if a match is not detected, re-executing the searching and determining instructions.

27. (original): The computer readable product of claim 19 wherein said computer executable instructions for adding a unique server identification code comprises computer executable instructions for:

if said client request does not comprise a session identification code;

creating a session for client requests received from the client issuing said client request;

assigning a session identification code to said session;

dispatching said client request to a particular server in said determined server group;

appending the server identification code of said particular server to said session identification code; and

communicating said session identification code including said server identification code, to the client that issued said client request.

28. (original): The computer readable product of claim 27 wherein said computer executable instructions for adding a unique server identification code further comprises computer executable instructions for:

when a client request is received that pertains to a particular server group and said client request comprises a session identification code, but said session identification code does not have appended thereto a server identification code corresponding to a server in said particular server group;

dispatching said client request to a server in said particular server group;

appending said server identification code corresponding to said assigned server to said session identification code; and

communicating said session identification code, including said newly appended server identification code, to the client that issued said client request.

29. (original): The computer readable product of claim 28 wherein said computer executable instructions for adding a unique server identification code comprises computer executable instructions for:

if a client request comprises a session identification code having appended

thereto a server identification code corresponding to a particular server in said determined server group, but said particular server is unavailable to service said client request;

dispatching said client request to a different server in said determined server group;

appending a server identification code corresponding to said different server to said session identification code; and

communicating said session identification code, including said newly appended server identification code, to the client that issued said client request.

EVIDENCE APPENDIX

No additional evidence is presented.

RELATED PROCEEDINGS APPENDIX

No related proceedings are presented.